Automatic Detection and Large-Scale Visualization of Trees for Digital Landscapes and City Models Based on 3D Point Clouds

Christoph OEHLKE[1], Rico RICHTER[1] and Jürgen DÖLLNER[1]

[1] Hasso Plattner Institute at the University of Potsdam,
Prof.-Dr.-Helmert-Str. 2-3, 14482, Potsdam, Germany
{christoph.oehlke,rico.richter,juergen.doellner}@hpi.de

Abstract

Vegetation objects represent one main compositional element of digital models of our environment required by a growing number of simulation, analysis, and visualization applications. However, a detailed representation of vegetation in 3D spatial models is generally not feasible due to the lack of up-to-date, object-based, and area-wide tree surveys and computational limits in data acquisition, storage, and visualization regarding vegetation. In this paper, we present an approach for automatic detection, categorization, and visualization of individual trees based on dense 3D point cloud analysis and efficient real-time rendering techniques such as instancing, adaptive tessellation, and vertex displacement. We have evaluated our approach for an urban area and a forest area with about 100 points/m², running real-time visualization on standard desktop hardware. The results indicate that this kind of automatic tree cadastre based on dense 3D point clouds is a practicable and cost-efficient approach to integrate area-wide, object-based vegetation models into virtual 3D landscape and 3D city models and, in particular, significantly enhance their visual appearance and their suitability for computational applications.

1 Introduction

Virtual 3D landscape models and 3D city models have become essential tools for various application domains such as landscape and urban planning, environmental monitoring, disaster and risk management, geodesign as well as homeland security. Generally, they combine multiple geo-spatial datasets to create digital representations of the real world, including models for buildings, streets, terrain surfaces and related objects (e.g., city furniture). Vegetation frequently covers a considerable amount of the terrain, represents a constituting and characteristic element, shapes the visual appearance, and serves as landmarks helping users to orientate themselves. Hence, vegetation significantly contributes to 3D spatial models and their applications in simulation, analysis, and visualization.

Traditionally, geo-spatial datasets for individual trees, called *tree cadastres*, are collected and maintained manually, which is an expensive and time-consuming process. As a remedy, these tree cadastres are often limited to selected regions of interest, such as roadsides. Moreover, manually created tree cadastres do not provide sufficient information about the real-world appearance of individual trees: Tree height and crown diameter, for example, are commonly omitted in a tree cadastre as it would require frequent updates due to vegetation growth. In addition, spectral characteristics of individual trees, such as typical leaf color,

Buhmann, E., Ervin, S. M. & Pietsch, M. (Eds.) (2015): Peer Reviewed Proceedings of Digital Landscape Architecture 2015 at Anhalt University of Applied Sciences. © Herbert Wichmann Verlag, VDE VERLAG GMBH, Berlin/Offenbach. ISBN 978-3-87907-555-3.

are rarely documented. However, these tree-specific parameters are essential for providing a realistic appearance of trees in the context of virtual 3D spatial models.

This poses two main challenges: The first challenge is the automatic, area-wide derivation and continuation of tree cadastres and corresponding parameters based on remote sensing data collected in regular intervals (e.g., once a year). In particular, dense 3D point clouds offer a computationally effective approach to automating detection, classification, and object-based representation. The second challenge is the efficient 3D rendering of large-area tree cadastres, since vegetation shows complex morphological characteristics, including branches, twigs, and leafs. A straightforward 3D rendering approach by placing an individual 3D tree model for each tree object is still not feasible, even in a long-term view, due to limited storage and processing capabilities of modern graphics processing units (GPUs). A number of specialized, optimized 3D rendering techniques (BAO et al. 2011; LIVNY et al. 2011) have been investigated but even they are far away from automatically generating individual 3D tree models for a real environment.

In our approach, we aim at automatically deriving tree objects and rendering these objects using templates that are configured by their individual characteristics. We identified the following requirements:

- **Automation of Data Collection:** The process of creating tree cadastres for arbitrarily large areas should be performed without any manual interaction, i.e., by automatic analysis of dense 3D point clouds together with aerial images.
- **Real-Time Rendering:** The rendering process should be capable of visualizing $>10^6$ trees at interactive frame rates, i.e., at least 15 frames per second.
- **Characteristic Appearance:** To achieve a high degree of realism, individual tree characteristics (e.g., tree height, crown diameter, typical leaf color, and main species) have to be mapped to each virtual tree model.

To address the issue of automatic vegetation detection (RICHTER et al. 2013) and tree cadastre generation, various approaches (JAKUBOWSKI et al. 2013) have been presented in recent years that automatically extract tree positions from 3D point clouds and aerial images, along with additional geometrical parameters (EDSON & WING 2011), such as tree height and crown diameter. Further, four-channel aerial images (RGBNIR) can be used to obtain information about tree species (ZHANG & HU 2012). To address the rendering issue, we use instancing as primary technique to render a large number of similar objects that are customized through object-specific rendering parameters (BAO et al. 2011).

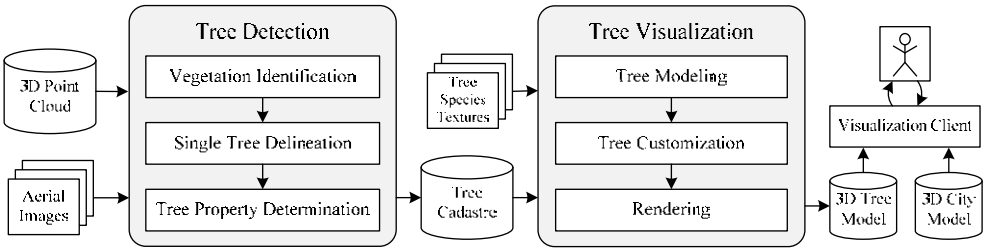


Fig. 1: Overview of system components and the corresponding processing pipeline

In this paper, we present a tree visualization approach for virtual 3D spatial models that combines automatic detection and real-time rendering of trees (Fig. 1). Based on a 3D point cloud and four-channel aerial images (RGBNIR), a tree cadastre is generated as a list of single trees (e.g., with a minimum height of 2m) along with tree-specific properties.

In particular, typical leaf colors are extracted from the aerial image using a novel approach based on shadow removal and statistics in HSV color space. Together with a pre-defined set of tree species textures, this tree cadastre is used for modeling, customizing and rendering trees for virtual 3D spatial models.

2 Tree Detection

Tree detection is a data processing task that automatically generates an object-based tree cadastre for a geographic area based on a dense 3D point cloud and corresponding four-channel ortho-images. Dense 3D point clouds result from remote sensing based on, e.g., LiDAR techniques or dense image matching. Tree detection comprises three consecutive steps: *vegetation identification*, *single tree delineation*, and *tree property determination*.

2.1 Vegetation Identification

As a preparation step, the dense 3D point cloud is separated into vegetation and non-vegetation points as described by RICHTER et al. (2013), based on analyzing the structure and topology of LiDAR or dense image matching point clouds. Subsequent processing stages operate on vegetation points with an elevation of at least 0.5 m above the ground.

2.2 Single Tree Delineation

Based on detected vegetation points, individual tree objects are separated from each other. Existing methods for single tree delineation can be classified into *image-based* and *point-based* approaches. Image-based approaches convert the 3D point cloud into a 2D height map and apply image processing algorithms on this height map, such as inverse watershed segmentation (REITBERGER et al. 2007) or edge detection (CHEN & ZAKHOR 2009). In contrast, point-based approaches operate on the 3D point data itself and use geometric approaches to delineate single trees, such as RANSAC methods (TITTMANN et al., 2011) or iterative, top-down classification based on horizontal spacing rules (LI et al., 2012).

Based on the approach of LI et al. (2012), we present a new, point-based tree delineation algorithm that requires only a single iteration over the 3D point cloud. During this iteration, each point in the 3D point cloud is assigned a *tree ID*, while all points belonging to the same tree are assigned the same tree ID. To take advantage of the horizontal spacing between treetops, the algorithm starts with the highest point in the 3D point cloud. For the current point p, a sub-procedure searches for the closest tree point q in the local neighborhood, i.e., the closest point that has already been assigned a tree ID, while the radius of this neighborhood linearly depends on the elevation of p above the ground. If such a point q exists in the local neighborhood, p belongs to the same tree as q and is therefore assigned the same tree ID. Otherwise, if there is no such point q in the local neighborhood, p is assigned a new tree ID, which is typically the case for the highest point within a tree.

2.3 Tree Property Determination

For each detected tree, *geometric* and *spectral* properties are derived that describe the individual tree characteristics. Geometric properties are extracted from the 3D point cloud and include *position*, *height* (elevation of the treetop relative to the ground), *crown diameter*, and *crown cover area* for each tree. Spectral properties are extracted from the corresponding aerial image of the tree crown and comprise *mean band ratios*, *mean NDVI*, and *mean channels in HSV color space*. The latter is used to identify the typical leaf color of individual trees to allow for realistic virtual representations. To minimize the effect of shadows, we apply a *shadow index* (SI), introduced by ONO et al. (2007), to identify shadowed regions on the tree crown and ignore these areas when computing spectral properties. As an additional application, spectral properties are useful to predict individual tree species.

3 Tree Visualization

The input of the tree visualization component is an object-based tree cadastre defining the position and characteristics of individual trees, and a set of 2D tree textures showing a tree from three different points of view (Fig. 2(a-c)).

To generate a 3D tree model, these textures are mapped onto separate, semi-transparent, intersecting quad geometries commonly denoted as *crossed billboards* (ZHANG et al. 2006) (Fig. 2(d-e)). Crossed billboards leverage the capabilities of modern texturing hardware to create visually complex tree models that can be rendered efficiently. However, when viewed at close range (Fig. 2(d-e)) or from an elevated position (Fig. 2(e)), crossed-billboard trees appear unrealistic due to the plain geometric structure, especially resulting in parallax artifacts. To increase the visual complexity of nearby trees, billboard planes are dynamically tessellated into a polygonal mesh that is deformed using vertex displacement. The resulting tree models (Fig. 2(f-g)) exhibit a more detailed, volumetric, and unstructured appearance leading to less parallax artifacts. Tessellation and vertex displacement are performed on the GPU and are only applied to tree models that are close to the viewer.

To render multiple trees efficiently, instancing is used as a means of minimizing the communication overhead between CPU and GPU. Moreover, trees outside the view frustum are excluded from rendering to increase the performance. To obtain a more diverse and realistic appearance, tree models are customized by their individual characteristics defined in the tree cadastre such as tree height and typical leaf color. Color transfer from the aerial image to the tree texture takes place in HSV color space and involves the following tasks:

- During tree detection, the mean hue (h_1), saturation (s_1), and value (v_1) channels are extracted from each tree crown in the corresponding aerial image.
- As a preprocessing step, the mean value (v_2) channel of all pixels belonging to leafs within a tree texture is extracted.
- At runtime, an individual tree texture is generated for each tree by using the mean hue (h_1) and saturation (s_1) from the corresponding aerial image and the original value channel from the tree species texture increased by $v_{shift} := v_1 - v_2$.

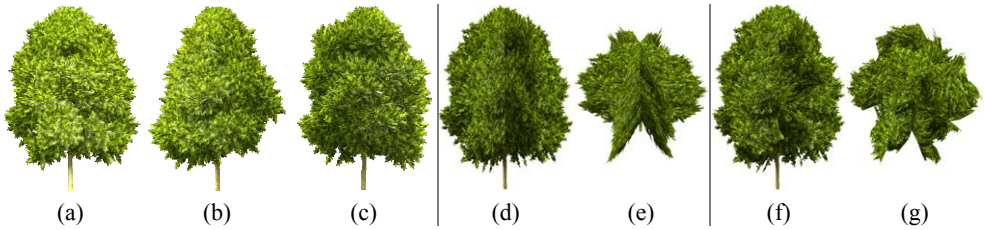


Fig. 2: Basic principle of instancing-based tree rendering. The input is a small set of 2D tree textures (a-c) used to form a crossed-billboard tree (d-e). To prevent parallax artifacts at close range, trees are adaptively tessellated (f-g).

4 Results and Discussion

The presented concepts have been implemented as a prototype application based on C++ and OpenGL. An evaluation of tree detection accuracy and visualization performance was performed on a desktop computer with an Intel Core i5 CPU at 3.30 GHz (4 cores and 4 threads), 8 GB of main memory, and an NVIDIA GeForce GTX 760 with 2 GB of video memory. The input 3D point cloud was generated through dense image matching and has a resolution of 100 points/m². The automatically generated tree cadastres were compared with manually classified reference data for two selected regions shown in Fig. 3. The suburban area (Fig. 3(a)) comprises a significant number of isolated, small- and medium-sized trees (up to 10 m) and some large, grouped trees. The urban forest (Fig. 3(b)) is mainly constituted by large, grouped trees (above 15 m). For three different height categories, the achieved precision, recall, and F-Measure were calculated as shown in Table 1. In both regions, small, isolated trees (below 10 m) are robustly detected, as indicated by the high recall of 97.4 % for the suburban area and 92.3 % for the urban forest. The least number of false positives occurred in the highest tree category (above 15 m), resulting in the highest precisions of 80.3 % for the suburban area and 91.1 % for the urban forest. In the suburban test area, the precision is mainly affected by image matching errors associated with city furniture, such as poles and traffic signs, resulting in a number of false positives.

Fig. 3: Overview of the two test areas: (a) Mostly isolated, urban trees; (b) urban forest

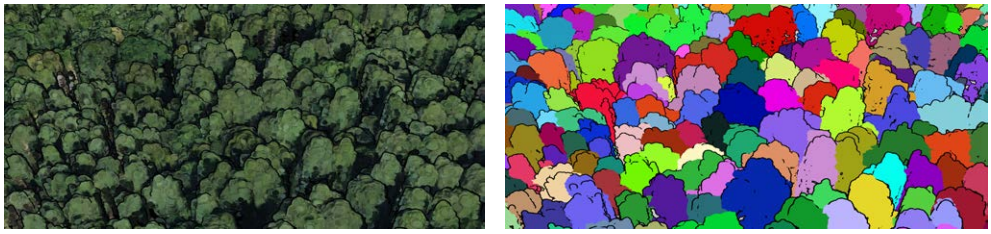
Table 1: Tree detection accuracy measured as precision, recall, and F-Measure

	Suburban area				Urban forest area			
Tree height (m)	**Count**	**Prec.**	**Recall**	**F-Meas.**	**Count**	**Prec.**	**Recall**	**F-Meas.**
[2;10)	214	71.9%	97.4%	82.7%	61	62.1%	92.3%	74.2%
[10;15)	115	69.6%	84.5%	76.3%	16	78.6%	84.6%	81.5%
[15;∞)	160	80.3%	72.1%	76.0%	341	91.1%	84.4%	87.6%
[2;∞)	489	73.7%	85.3%	79.1%	418	86.0%	85.3%	85.6%

Fig. 4 shows a visualization of single tree results delineation for a dense forest area. Most tree crowns were separated correctly, as indicated by different colors, but the automatic approach fails at separating tree crowns that are tightly interconnected, creating some degenerated large segments. This is especially due to the limited ability of 3D point clouds generated by dense image matching to describe the complete, three-dimensional structure of dense vegetation. For visualization purposes, however, the uncorrected tree cadastre may already be sufficient to derive a plausible and realistic model of the overall vegetation.

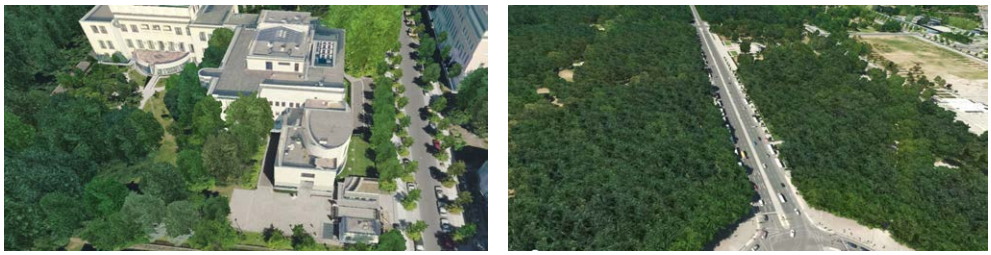
Fig. 5 shows two visualizations of an automatically generated tree cadastre for an urban area in the context of a virtual 3D city model. Fig. 5 (a) demonstrates how individual tree models are scaled to their individual height, as indicated by the group of old trees on the left and the small roadside trees on the right. Moreover, different leaf colors are observable that were extracted from the aerial image and mapped onto the individual tree crowns. Fig. 5 (b) shows an urban forest from a distant view.

Tree rendering performance has been evaluated on a synthetic, randomly generated tree cadastre, a window resolution of 1920×1080, and two fixed camera orientations. During the test, the amount of trees was gradually increased to investigate the impact on the achieved frame rate in *frames per second* (FPS), as shown in Fig. 6. Three different render settings were compared to each other: First, all trees were rendered with full tessellation (294 triangles per tree) and without culling. Second, adaptive tessellation was enabled to reduce the geometric complexity for distant trees to a minimum of 6 triangles per tree. Third, adaptive tessellation was supplemented with individual tree culling, further reducing the rendering overhead for trees outside the view frustum. The first setting shows similar performance for both camera orientations, achieving a frame rate of 10 FPS for up to 60k trees. Our adap-



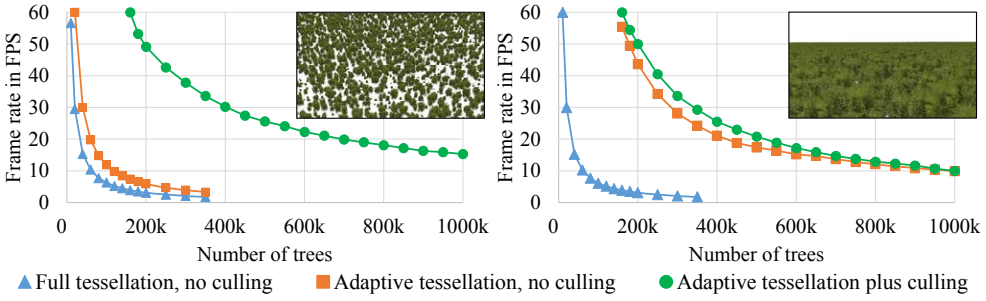
(a) (b)

Fig. 4: Visualization of automatic single tree delineation results for a dense forest area (a). Each identified tree is visualized with a random color (b).



(a) (b)

Fig. 5: Automatically generated tree models visualized by a virtual 3D city model



(a) (b)

Fig. 6: Tree scene used for performance evaluation (a) and frame rate depending on the number of rendered trees (b)

tive tessellation technique increases the performance for scenarios with flat viewing angles and a large amount of visible trees (Fig. 6 (b)), while our culling approach improves the performance for scenarios with close views (Fig. 6 (a)). Hence, up to 500k trees can be rendered in real time, whereas up to one million trees can be rendered at interactive frame rates on standard consumer hardware.

5 Conclusions and Outlook

To facilitate automatic tree detection based on dense 3D point clouds, we use topology analysis to identify vegetation and develop a new, iterative algorithm for single tree delineation. The algorithm does not assume a specific data acquisition method, i.e., it can be used for airborne LiDAR as well as for dense image matching. Moreover, there is no need for analyzing test data sets, applying machine learning, or implementing heuristics, as dense 3D point clouds offer sufficient geometric and topologic information. Due to these properties, the presented approach for automatically deriving tree cadastres based on dense 3D point clouds proves to be practicable, robust, and cost-efficient in terms of processing time and data requirements. We further contribute a corresponding real-time 3D rendering technique. Therefore, tree detection derives tree-specific properties with a particular focus on the needs of visualization. Our rendering approach uses simple, traditional crossed bill-

boards together with a novel GPU shader for adaptive tessellation and vertex displacement applied to these billboards. To achieve interactive frame rates even for large numbers of trees as they occur for large-scale city or landscape models, efficient instancing and instance culling are applied. As a key quality of our rendering approach, the appearance of each tree model is tailored to its individual key visual properties derived from the aerial photography. This way, visualized vegetation objects reflect key visual characteristics and, therefore, ensure that vegetation can be easily recognized in the corresponding virtual 3D model. The results of our test data sets indicate that the algorithm detects most urban trees even if the crown is partly shadowed. Furthermore, the tree visualization results demonstrate that the use of remote sensing data and modern rendering techniques is an effective means of integrating tree models into 3D spatial models. The presented approach complements workflows in today's GIS and GDIs, efficiently deriving and integrating area-wide, object-based vegetation models into virtual 3D landscape and 3D city models.

Remaining challenges are the reliable identification of vegetated roofs and the separation of tightly interconnected tree crowns. Moreover, the visual quality of the derived tree models should be further evaluated in comparison to traditional, explicit tree models with respect to perceivable differences and visible artifacts due to geometry simplification. In addition, new tree color styles could be used to convey an impression of seasons: For example, a fall scenario could be created by adjusting leaf colors to yellow and red.

Acknowledgements

This work was funded by the Federal Ministry of Education and Research (BMBF), Germany within the InnoProfile Transfer research group ”4DnD-Vis” (www.4dndvis.de). We would like to thank virtualcitySYSTEMS for providing datasets.

References

BAO, G., MENG, W., LI, H., LIU, J. & ZHANG, X. (2011), Hardware instancing for real-time realistic forest rendering. SIGGRAPH Asia Sketches, Article 16.

CHEN, G. & ZAKHOR, A. (2009), 2D Tree Detection in Large Urban Landscapes Using Aerial Lidar Data. 16th IEEE International Conference on Image Processing (ICIP), 1693-1696.

EDSON, C. & WING, M. G. (2011), Airborne Light Detection and Ranging (LiDAR) for Individual Tree Stem Location, Height, and Biomass Measurements. Remote Sensing, 3 (11), 2494-2528.

JAKUBOWSKI, M. K., LI, W., GUO, Q. & KELLY, M. (2013), Delineating Individual Trees from Lidar Data: A Comparison of Vector- and Raster-based Segmentation Approaches. Remote Sensing, 5 (13), 4163-4186.

LI, W., GUO, Q., JAKUBOWSKI, M. K. & KELLY, M. (2012), A New Method for Segmenting Individual Trees from the Lidar Point Cloud. Photogrammetric Engineering & Remote Sensing, 78 (1), 75-84.

LIVNY, Y., PIRK, S., CHENG, Z., YAN, F., DEUSSEN, O., Cohen-Or, D. & Chen, B. (2011), Texture-Lobes for Tree Modelling. ACM Trans. Graph., 30 (4), Article 53.

REITBERGER, J., HEURICH, M., KRZYSTEK, P. & STILLA, U. (2007), Single Tree Detection in Forest Areas With High-Density Lidar Data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36, 139-144.

RICHTER, R., BEHRENS, M. & DÖLLNER, J. (2013), Object class segmentation of massive 3D point clouds of urban areas using point cloud topology. International Journal of Remote Sensing, 34 (23), 8408-8424.

TITTMANN, P., SHAFII, S., HARTSOUGH, B. & HAMANN, B. (2011), Tree Detection and Delineation from LiDAR point clouds using RANSAC. SilviLaser 2011.

ZHANG, K. & HU, B. (2012), Individual Urban Tree Species Classification Using Very High Spatial Resolution Airborne Multi-Spectral Imagery Using Longitudinal Profiles. Remote Sensing, 4 (6), 1741-1757.

ZHANG, Y.-K., TEBOUL, O., ZHANG, X.-P. & DENG, Q.-Q. (2006), Image based real-time and realistic forest rendering and forest growth simulation. PMA '06 Plant Growth Modeling and Applications, 323-327.